

METHOD AND SYSTEM OF LANGUAGE DETECTION

Shamim A. Alpha

FIELD OF THE INVENTION

[0001] The invention relates to the computerized language detection arts. It finds particular application to a method and system of determining the language of a text document. It will be appreciated that the invention will also find application to information retrieval engines or other document search engines.

BACKGROUND OF THE INVENTION

[0002] With the expansion of global computer networks, users can search for and gather documents from many sources many of which maybe unknown and often unmanaged sources. The language of documents found is not necessarily known a priori.

[0003] Traditional language detection techniques try to answer what are the chances that a document is in a given language. This answer is computed in isolation irrespective of what other languages the document could possibly be. Prior methods try to determine if documents in a given language will also have similar N-gram or small word distribution.

[0004] If the languages in a candidate list are significantly different from each other, the traditional technique works reasonably well. However, when languages are very similar to each other, many false positives are obtained.

[0005] The present invention provides a method and system for detecting the language of an unknown text that cures the above problems and others.

SUMMARY OF THE INVENTION

[0006] According to one embodiment of the present invention, a system for automatically determining a language of a document from a set of candidate of languages is provided. The system includes logic that sets a negative assumption value indicating the document is not one of the candidate languages. An extractor then extracts a character string from the document. A language analyzer determines a probability value that the character string does not belong to the candidate languages and the negative assumption value is adjusted based on the probability value. The language analyzer determines that the document is one language of the candidate languages when the negative assumption value passes a threshold value.

[0007] In accordance with a more limited aspect of the present invention, the system further includes an information retrieval engine that retrieves documents in response to a search request. The documents retrieved being analyzed by the language analyzer to determine their language.

[0008] In accordance with another embodiment of the present invention, a method of determining a language of a document from a set of candidate languages is provided. A null hypothesis is set to a true value for each candidate language indicating the document is not in the candidate language and a false value is set. A text string is extracted from the document and a contrary probability is determined for each candidate language that the text string does not belong to the candidate language. The null hypothesis is then adjusted for each candidate language with the contrary probability corresponding to the candidate language. The document is determined to be one language from the candidate languages when the null hypothesis for the one language is disproved by approaching the false value.

[0009] In accordance with another embodiment of the present invention, a process of determining that a document is in a selected language is provided. The process includes

setting a probability assumption indicating that the document is not in the selected language. A character string is extracted from the document. The probability assumption is then disproved based on a contrary probability that the character string does not belong to the selected language such that if the contrary probability fails to support the probability assumption, then the document is determined as being in the selected language.

[0010] One advantage of the present invention is that it increases the accuracy of determining the language of a document.

[0011] Another advantage of the present invention is that it does not determine probabilities that characters belong to a language based on an isolated view of only that language. Rather, it determines probabilities that characters belong to a language based on how those characters belong to all candidate languages.

[0012] Still further advantages of the present invention will become apparent to those of ordinary skill in the art upon reading and understanding the following detailed description of the preferred embodiments.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] In the accompanying drawings which are incorporated in and constitute a part of the specification, embodiments of the invention are illustrated, which, together with a general description of the invention given above, and the detailed description given below, serve to exemplify the principles of this invention.

[0014] Figure 1 is an exemplary overall system diagram of a language analyzer in accordance with the present invention;

[0015] Figure 2 is an exemplary diagram showing how probability data is generated in accordance with the present invention;

[0016] Figure 3 is an exemplary system using the present invention; and

[0017] Figure 4 is an exemplary methodology of determining the language of a document in accordance with the present invention.

DETAILED DESCRIPTION OF ILLUSTRATED EMBODIMENT

[0018] The following includes definitions of exemplary terms used throughout the disclosure. Both singular and plural forms of all terms fall within each meaning:

[0019] “Document”, as used herein, includes but is not limited to an electronic document, a web page or any object having text.

[0020] “Software”, as used herein, includes but is not limited to one or more computer executable instructions, routines, algorithms, modules or programs including separate applications or ones from dynamically linked libraries for performing functions as described herein. Software may also be implemented in various forms such as a servlet, an applet, a stand-alone program including a server based application and a user based application, a plug-in or other type of application. Software may also be stored on various computer readable mediums such as disk, CD, tape, memory and can be downloadable.

[0021] “Logic”, as used herein, includes but is not limited to hardware, software and/or combinations of both to perform one or more functions.

[0022] “Network”, as used herein, includes but is not limited to the internet, intranets, Wide Area Networks (WANs), Local Area Networks (LANs), and transducer links such as those using Modulator-Demodulators (modems).

[0023] “Internet”, as used herein, includes a wide area data communications network, typically accessible by any user having appropriate software.

[0024] “Intranet”, as used herein, includes a data communications network similar to an internet but typically having access restricted to a specific group of individuals, organizations, or computers.

[0025] Illustrated in **Figure 1** is an exemplary system diagram in accordance with the present invention. The system is embodied as software and includes a language analyzer **100** that determines the language of text from a document **105**. In one embodiment, the language analyzer **100** is a component within an information retrieval engine **110** or is called by the engine **110** when a document is retrieved or identified by the engine. The information retrieval engine **110** is, for example, Oracle Text, made by Oracle Corporation or any type of search engine or document retrieval system as is known in the art.

[0026] Briefly describing the system, a text extractor **115** searches for and extracts a portion of text from the document **105** for analysis. For a given set of candidate languages, a probability logic **120** sets a null hypothesis representing “true” that negatively assumes the document **105** is not in any of the given languages. The language analyzer then tries to refute the assumption by trying to prove the null hypothesis is “false”. In other words, the negative assumption starts with a probability estimate for each candidate language of a 100% value that the document is not English, is not French, is not Spanish, etc. Predetermined probability data **125** (described below) is used to adjust the probability estimate for each language based on a probability that the extracted text does not belong to the language. Once the probability estimate for one of the languages falls below a selected threshold, its negative

assumption fails. Thus, the document is most likely in that language because the attempt to prove the contrary has failed. At that point, the system determines that too much evidence contrary to the null hypothesis is found such that the probability estimate supporting the null hypothesis is minuscule. If the threshold is not passed, iteration logic within the language analyzer **100** repeats the process with another extracted portion of text.

[0027] Illustrated in **Figure 2** is a system and process for generating the probability data **125** that is formed as, for example, one or more data tables or databases. The probability data is generated for a selected set of languages, for example, European languages. To describe the invention, English, French and Spanish are the selected languages but any number can be selected. For each language, a sample set of documents are selected for analysis, e.g. English documents **200**, French documents **205** and Spanish documents **210**. Each sample set of documents may include hundreds or thousands of documents or more. When the probability data **125** is completed, it will contain probability values that a string of text occurs within a language based on its occurrences in all the languages.

[0028] With reference to the English documents **200**, a software text scanner **215** searches the text of each document and extracts sequential strings of characters. Each extracted string has a predetermined size, for example, three characters. An occurrence counter **220** includes logic that counts the number of occurrences of each extracted string within the set of English documents **200**. The number of occurrences for each extracted string is then stored in an occurrence table **225** for English. For example, if an extracted string “ace” occurs five (5) times, that value is stored in the table **225**. This process repeats for other three letter strings throughout the set of English documents **200** using iteration logic.

[0029] With further reference to **Figure 2**, the French documents **205** and Spanish documents **210** are also inputted to the text scanner **215**. A corresponding French occurrence

table **230** and Spanish occurrence table **235** are generated having occurrence values of three letter strings. Using “ace” as an example, it may have occurred two hundred (200) times in the French documents and fifty (50) times in the Spanish documents. Those values are then stored in the French table **230** and the Spanish table **235**, respectively.

[0030] A normalizer **240** includes logic that normalizes each occurrence value for each string based on the same string’s occurrence values in all the selected languages, e.g. from all the occurrence tables **225**, **230** and **235**. In this manner, the system does not determine probabilities that characters belong to a language based on an isolated view of only that language. Rather, it determines probabilities that characters belong to a language based on how those characters belong to all candidate languages. The normalized value for a string becomes the probability that a document is in a language given that the string is found in the document. Each occurrence value is normalized as shown in Equation (1):

$$P_{iEng}(xxx) = \frac{M_{iEng}(xxx)}{M_{iEng}(xxx) + M_{iFren}(xxx) + M_{iSpan}(xxx)} \quad (1)$$

[0031] $M_{iEng}(xxx)$ is the number of occurrences of a string “xxx” found in the English occurrence table **225**. $M_{iFren}(xxx)$ and $M_{iSpan}(xxx)$ are the number of occurrences of the same string “xxx” found in the French and Spanish occurrence tables **230** and **235**, respectfully. In other words, based on the occurrence tables, if “ace” occurred 5 times in English, 200 times in French and 50 times in Spanish, using maximum likelihood estimate, the normalized probability $P_{iEng}(ace)$ that a document is English given that “ace” is found is $5/(5+200+50)$ or approximately .02 (2%). Normalized probabilities are also determined for French and Spanish entries of “ace” as $P_{iFren}(ace) = 200/(5+200+50)$ or approximately .78 (78%) and $P_{iSpan}(ace) = 50/(5+200+50)$ or approximately .20 (20%). Using an iterative process, the normalizer **240** computes these normalized probabilities for all occurrence values from tables **225**, **230** and **235** and stores them as the probability data **125**. Thus, the probability data **125**

associated with a string of text “xxx” is influenced by the occurrences of “xxx” in all selected languages.

[0032] The probability data 125 may be configured in many ways as will be appreciated by one of ordinary skill in the art. For example, each string can be used as a key index for each candidate language. In this manner, the probability data 125 is searchable by the string to retrieve its corresponding probability value. In another embodiment, each string may be used to obtain an index value, by using a hashing function or the like, that associates the string to a value in the probability data 125 for each candidate language.

[0033] Illustrated in **Figure 3** is an exemplary system diagram showing an application of one embodiment the invention. A computer system 300 receives document search requests from one or more user systems 305 connected thereto via a network 310 or by direct connection. The computer system 300 executes an information retrieval engine 315 to locate candidate documents that match the search request to a selected degree as is known in the art. Since one or more of the candidate documents may be retrieved from unknown sources, the language analyzer 100 is executed to determine the language of any document using the probability data 125 as described above.

[0034] Generally, the computer system 300 may take many forms from a configuration including a variety of processing units networked together to function as a integral entity, to a single computer, e.g. a personal computer, operational in a stand-alone environment. The present invention can be embodied in any of these computer system configurations.

[0035] Illustrated in **Figure 4** is an exemplary computer-implemented methodology of the language analyzer 100 that detects the language of a document in accordance with the present invention. The blocks shown represent functions, actions or events performed therein. It will be appreciated that computer software applications involve dynamic and

flexible processes such that the illustrated blocks can be performed in other sequences different than the one shown. It will also be appreciated by one of ordinary skill in the art that the software of the present invention may be implemented using various programming approaches such as procedural, object oriented or artificial intelligence techniques.

[0036] With reference to **Figure 4**, a document is retrieved in response to a search request (block **400**). It will be appreciated that the document might not be physically received but rather its location can be simply identified. A negative assumption is set (block **405**) for each candidate language that the document is not that language and the analysis attempts to prove the negative assumption wrong. If the negative assumption fails for one language, then the document is identified as being in that language.

[0037] The negative assumption sets a probability estimate P_{doc} at “1” (100% true value) for each language that the document is not in that language:

$$P_{\text{doc}}(\text{not English}) = 1$$

$$P_{\text{doc}}(\text{not French}) = 1$$

$$P_{\text{doc}}(\text{not Spanish}) = 1$$

[0038] The document is then searched and a portion of text **T** is extracted (block **410**). The text, in this case, is a three character string since the probability data **125** was generated based on three letter strings. If a different string length is used for the probability data **125**, the same length would be extracted here. To simplify the explanation, assume that the first extracted string is “ace” as used above. The probability data **125** is then searched for an entry matching “ace” in each of the languages (block **415**). The probability data **125** returns a probability value for each language that reflects the probability that the document is that language given the occurrence of the string “ace”. The probabilities, as taken from the above example, are as follows:

$$P_{\text{Eng}}(\text{ace}) = .02$$

$$P_{\text{Fren}}(\text{ace}) = .78$$

$$P_{\text{Span}}(\text{ace}) = .20$$

[0039] These probabilities are then reversed or made contrary (e.g. subtracted from 1) to obtain a probability that the document is not in the language given the occurrence of the string “ace” (block 420). The contrary probabilities become:

$$P_{\text{notEng}}(\text{ace}) = .98$$

$$P_{\text{notFren}}(\text{ace}) = .22$$

$$P_{\text{notSpan}}(\text{ace}) = .80$$

[0040] The above contrary probabilities state that given the occurrence of the string “ace” there is a 98% probability that the document is not English, a 22% probability that the document is not French, and an 80% probability that the document is not Spanish. It will be appreciated that the probability data 125 can be generated to contain the contrary probabilities for each string, thus, eliminating this step here.

[0041] The probability estimate P_{doc} is then adjusted for each language based on the contrary probabilities of the current extracted string “ace” (block 425). In the below equations, it will be understood that the values to the right of the “=” represent a value “before” execution of the equation and the values to the left of the “=” represent the value “after” execution as known in computer processing. The adjusted values are found by:

$$P_{\text{doc}}(\text{not English}) = P_{\text{doc}}(\text{not English}) * P_{\text{notEng}}(\text{ace}) \quad (2)$$

$$P_{\text{doc}}(\text{not French}) = P_{\text{doc}}(\text{not French}) * P_{\text{notFren}}(\text{ace}) \quad (3)$$

$$P_{\text{doc}}(\text{not Spanish}) = P_{\text{doc}}(\text{not Spanish}) * P_{\text{notSpan}}(\text{ace}) \quad (4)$$

[0042] By multiplying the probability estimate with the contrary probability for the current string, the probability estimate will stay near “1” (meaning the negative assumption is true and the document is probably not in that language) or approach “0” (meaning the negative assumption is false and the document is probably in that language). With $P_{\text{doc}}(\text{not language})$ initially equal to “1”, after a first adjustment, the probability estimate P_{doc} becomes:

$$P_{\text{doc}}(\text{not English}) = .98$$

$$P_{\text{doc}}(\text{not French}) = .22$$

$$P_{\text{doc}}(\text{not Spanish}) = .80$$

[0043] After the first iteration, the negative assumption for French is closer to failing (e.g. approaching “0”) than the other languages. Thus at this point, there is a greater probability that the document is French. The analysis continues by comparing the current probability estimates to a selected threshold value (block 430) that is near “0” (a false value), for example 0.001. Of course, any threshold value may be chosen depending on the accuracy desired (e.g. 0, 0.01, 0.5, 0.1, etc.). Once the probability estimate P_{doc} for one language falls below the threshold, the document is determined to be in that language and the analysis stops (block 435). If the threshold is not passed, the analysis continues by extracting the next three letter string from the document and repeating the process (block 440).

[0044] Assuming that the next extracted string is represented by “xxx”, the process performs a second iteration. With reference to Table (1) below, we assume that “xxx” has the shown contrary probabilities for not being in a language which are determined from the probability data 125. As described above, the probability estimate P_{doc} is adjusted by the current contrary probabilities for “xxx” by multiplying them together. After the second iteration, the probability estimate P_{doc} for Not French is quickly approaching zero which means the document is probably French. The process continues if the threshold value is not passed. In this example, the system is assuming that occurrences of “ace” and “xxx” are independent of each other. Due to this assumption and initial conditions $P_{\text{doc}}(\text{not language}) = 1$, $P_{\text{doc}}(\text{not language})$ will not strictly reflect the probability of the document not being in that language. Instead, $P_{\text{doc}}(\text{not language})$ will be useful for comparison between different languages. Hence at any point, $P_{\text{doc}}(\text{not language X}) < P_{\text{doc}}(\text{not language Y})$ implies that the document is more likely to be in language X than in language Y.

Table (1):

Iteration	Not English	Not French	Not Spanish
1. Prob(ace)	.98	.22	.80
2. Prob(xxx)	.99	.30	.70
P_{doc}	.97	.066	.56

[0045] Combining the probabilities of each iteration is based on a principle that there is a relationship between phrases in a given language. For example, if a document is known to be in English and the phrase “you” occurs, then there is a greater probability that the phrase “ion” will also occur since it is common in English. This relationship principle is used by the analysis as described above when trying to prove or disprove the negative assumption. Thus when “ace”, which has a low probability (0.22) of not being French, is found in an unknown document, the failure of the negative assumption that the document is not French is further supported by subsequently finding another string “xxx” which also has a low probability (0.3) of not being French.

[0046] The present invention determines the language of an unknown document by using language specific characteristics of a language as compared to other candidate languages. As described above, probabilities that characters belong to a language are determined based on how those characters belong to all candidate languages rather than taking an isolated view of only one language. In this manner, the present invention provides increased robustness and reduces the probability of false positives.

[0047] While the present invention has been illustrated by the description of embodiments thereof, and while the embodiments have been described in considerable detail, it is not the intention of the applicants to restrict or in any way limit the scope of the appended claims to such detail. Additional advantages and modifications will readily appear to those skilled in the art. For example, the language analyzer may be a feature within a search engine or may be a separate program. Additionally, any string length can be used for the extracted string and any number of candidate languages can be used. The probability data may contain the contrary probabilities in order to eliminate the subsequent determination step. Therefore, the invention, in its broader aspects, is not limited to the specific details, the representative apparatus, and illustrative examples shown and described. Accordingly, departures may be made from such details without departing from the spirit or scope of the applicant's general inventive concept.